Mapping Language for Information Integration

Haridimos Kondylakis[1], Martin Doerr[1], Dimitris Plexousakis[1]

[1] Institute of Computer Science, FORTH-ICS
P.O. Box 1385, GR 71110, Heraklion, Crete, Greece
{kondylak, martin, dp}@ics.forth.gr

Abstract. An essential matter in heterogeneous database integration is the mapping process. In this report we present a mapping language for information integration under a common knowledge representation model (LAV approach). Based on particular requirements for a sufficient quality of information integration to be achieved, we present the most common cases of heterogeneity encountered in a wide sample of cases from museum collections, archeological, medical and genomic data. and reflect the individual investigated domain. We propose a specific mapping annotation format that is capable to capture all those cases. We assume that the level of detail of this format is sufficient to produce complete mediation of data transformation algorithms without further input from the domain experts. This assumption has to be verified in further work.

1. Introduction

Data Integration is one of the key problems for the development of modern information systems. The exponential growth of the web and the extended use of database management systems has brought to the fore the need for seamless interconnection of diverse and large numbers of information sources. In order to provide uniform access to heterogeneous and autonomous data sources, complex query and integration mechanisms have to be designed and implemented.

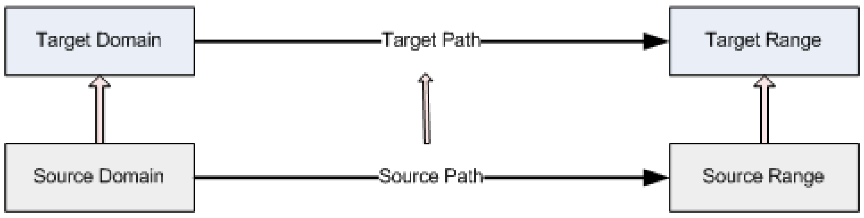

Fig. 1.: The basic mapping schema

An essential matter in heterogeneous database integration is the mapping process. We definition the mapping of two schemata as a sufficient specification to transformation of each instance of schema 1 into an instance of schema 2 with the same meaning" as shown in figure 1. The definition should be independent of particular instances. Mapping should allow for implementing an automatic transformation algorithm for all instances of schema 1 into instances of schema 2, only following the specification of the transformation.

In this work we assume that the target schema is an ontology or a knowledge representation model that uses nodes, links, properties, multiple ISA's and multiple instantiations to describe an appropriate domain of interest. The target schema does not enforce cardinality constraints on links, so we don't have to deal with heterogeneity concerning cardinalities. Moreover, we assume that the source schemata can be described using entity-relationship or XML models.

We suppose that the mapping definitions are produced manually or semi-automatically by a domain expert, possibly assisted by an IT expert. A suitable part of an ontology in a suitable encoding can be used or interpreted as Target Schema. In this report we use as example target schema the CIDOC Conceptual Reference Model that provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation (Crofts et al. 2005). Whereas the CIDOC CRM was created for information from cultural heritage, it is adequate to model other domains as well.

2. Cases of Heterogeneity that mapping should cover

The process of creating the mapping rules is not a straightforward process since multiple conditions and cases may exist. The problem of defining mappings between arbitrary schemata can not be solved in all cases. Moreover instances of schemata may not follow the intended meaning of the source schema and thus cause exceptions. However in a given domain and assuming certain qualities of the target schema, the cases of heterogeneity are normally quite limited.

In order to overcome those problems we suggest defining a mapping mechanism that will cover the most common cases and to extend the mechanism on demand – may be by inserting custom functions, pre- or post-processing - to cover the cases we have missed so far. Those most common cases were collected from museum collections, archeological, medical and genomic data and models. We believe that the most common cases in these domains reflect as well the general most common cases.

The mapping mechanism should be intuitive enough so that the domain expert can understand it, use it, or at least verify it. In order to do that we should examine carefully the most common cases of heterogeneity between the Source and Target Schema in our application context.

Our model can be used for mapping from XML and Relational Databases to an ontology defined in RDF-OWL or similar formats. Moreover this model can be used in simple object-oriented databases too. For relational databases, we interpret their schema as Semantic model. In order to do that we

- Interpret tables, columns as entities
- Interpret complete records as entity instances
- Interpret fieldnames both as relationships and entities
- Interpret field contents as entity instances.

Each field is interpreted as class-role-class (c-r-c) in the sequence called source path, the whole schema is decomposed in c-r-c's and each c-r-c is mapped individually to the target schema. In order to have an efficient mapping we need to define.

a) The mapping between the Source Domain and the Target Domain.
b) The mapping between the Source Range and the Target Range.
c) The proper Source Path.
d) The proper Target Path.
e) The mapping between Source Path and Target Path
f) In some cases, we may need to combine paths sharing the same instances

To the best of our knowledge, no other mapping language or mechanism provides such definitions, and most of them assume that some of those are implicitly defined by procedural code. We argue that all of the previous definitions should be explicitly defined in order to have efficient mapping rules.

Below, we present the most common cases of heterogeneity encountered. We have to note that the CIDOC CRM does not enforce cardinality constraints in the target schema.

Case 1. Introducing an intermediate node.

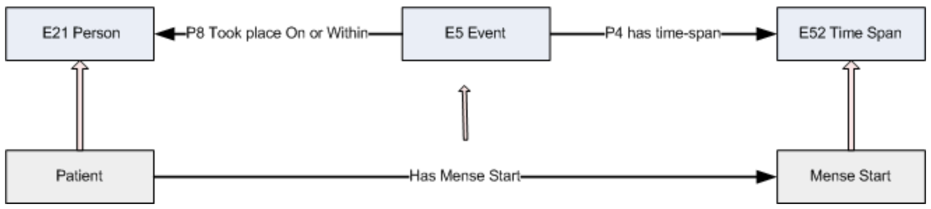

Fig. 2. Introducing an intermediate node

In this case, an intermediate node should be introduced in order to define precisely the whole path that needs to be mapped to the source path. This is because in the source model, it is common to compact a large path into a single relation (usually events) when no information needs to be stored in the intermediate nodes. Those intermediate nodes are necessary when other schemata have information that relates the intermediate node. This is implemented in the proposed mapping format by the "*internal_link*" and "*internal_path*" tags than can exist within a "*target_path*" tag that provides us the capability to have many intermediate link and paths.

Case 2. Compound Contraction

Frequently observed mainly in addresses, species names, coordinates etc, is the Compound contraction. In this case, several classes in the source schema are parts of one identifier for one real-life thing, one class in our target schema. There must be a way to declare that all those classes are parts of the same class in out target schema (Gruber 1993). This is implemented in the proposed mapping format by the attribute "*compound_on*" of the "*combined_links*" tag.

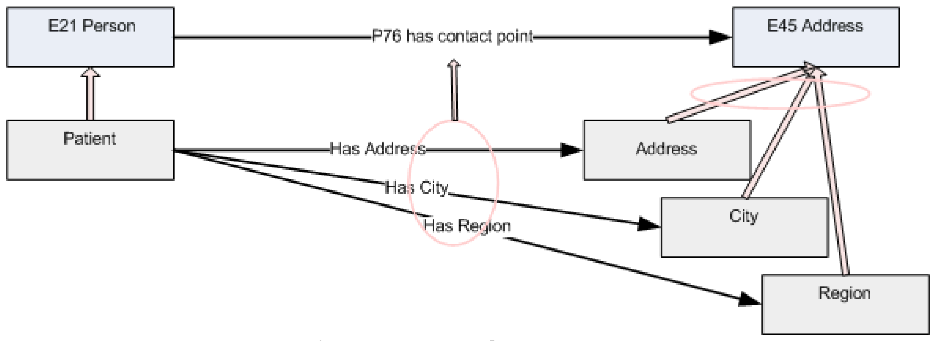

Fig. 3. Compound Contraction

Case 3. Parallel to Nested

There are cases, where in the source schema, a class A is related to other classes Bi, and those relations imply causal connection between some of the related classes Bi, rather than between A and Bi, as stated in the schema. This denormalized form is typically idiosyncratic to one source schema. Therefore those connections must be made explicit in order to insure interoperability with information from other sources. For example as we can see in previous figure, in the source model. “Hybridization” is related to “*Labeled Extract Quantity*”. However “*Labeled Extract Quantity*” is a class that should be related to a “*Labeled Extract*” since in reality it is it’s attribute.

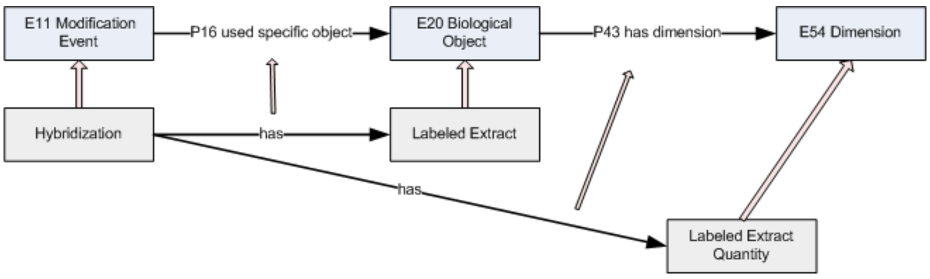

Fig. 4. Parallel to nested

Case 4. Parallel to Intermediate Parallel

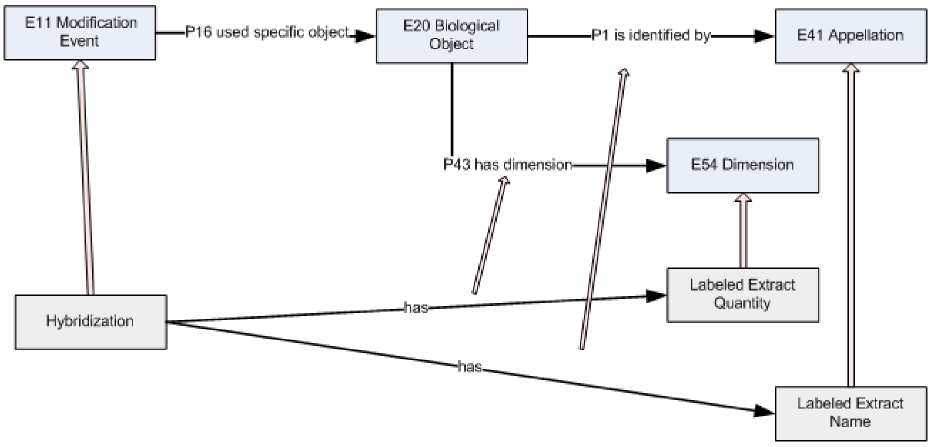

Fig. 5. Parallel to Intermediate Parallel

This case only extends the previous one: Two relations in the source schema mapped to a path in the target schema with the same intermediate node. Note that not only the class of the intermediate node is the same but also the instance by which the intermediate node has to be instantiated is the same for the mapping of both paths.

Case 5. Same instance participates in multiple mappings

Therefore, we need a general mechanism to define that the same instance of a class in a mapping rule appears in multiple mappings. This is implemented in the proposed mapping format by the attribute "*joined_on*" of the "*combined_links*" tag and by using several "*link_maps*" within the "*combined_links*" tag.

Case 6. Conditions

Finally we need a mechanism to define the following simple conditions the mapping depends on:

a)That an attribute of an instance is equal to a term or constant

e.g if instance.Attribute=" … "

b) That an attribute of an instance is a term a subsumed by another term b

e.g if instance.Attribute $\subset$ term b

c) That an instance of the attribute exists or not.

Those most common conditions cases described here, are confirmed from the MIDAS (a manual and data standard in monument inventories developed by FISH) mapping effort using CIDOC CRM. This is implemented in the proposed mapping format by the tags "*src_path_condition*", "*target_path_condition*", "*src_domain_condition*", "*target_range_condition*" and the "*value_binding*" attribute of the "*internal_link*" tag.

3. Proposing a Mapping Annotation Format

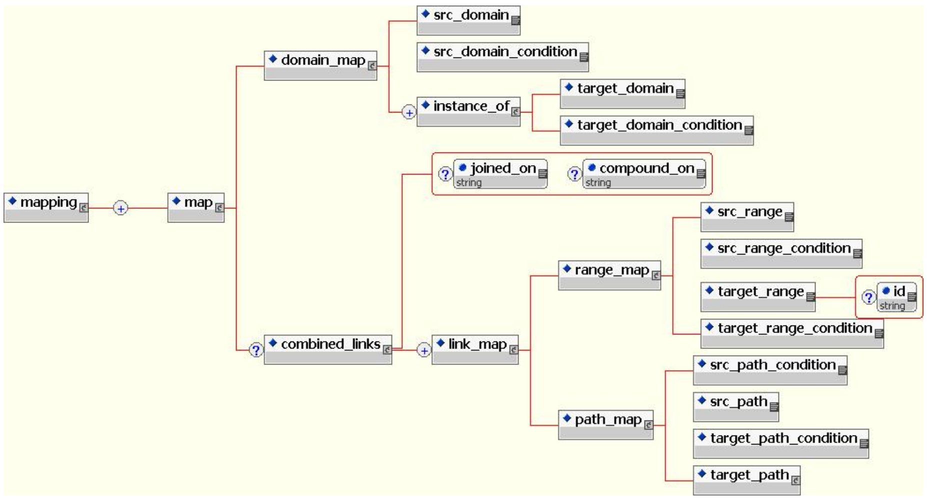

Fig. 6. The schematic DTD of the mappings file

The basic mapping annotation format and the corresponding DTD is shown in figure 6 and 7.

Specifying Mapping Conditions

Moreover, we have to specify those mapping conditions. In those conditions, AND, OR, NOT operators should be available. The syntax proposed is shown in the figure 8.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!ENTITY % unnamed "">
<!ELEMENT mapping (map)+>
<!ELEMENT map (domain_map , combined_links?)>
<!ELEMENT domain_map (src_domain , src_domain_condition ,
target_domain , target_domain_condition)>
<!ELEMENT range_map (src_range , src_range_condition ,
target_range , target_range_condition)>
<!ELEMENT path_map (src_path_condition , src_path ,
target_path_condition , target_path)>
<!ELEMENT src_domain (#PCDATA)>
<!ELEMENT src_domain_condition (#PCDATA)>
<!ELEMENT target_domain (#PCDATA)>
<!ELEMENT target_domain_condition (#PCDATA)>
<!ELEMENT src_range (#PCDATA)>
<!ELEMENT src_range_condition (#PCDATA)>
<!ELEMENT target_range (#PCDATA)>
<!ATTLIST target_range  id CDATA  #IMPLIED >
<!ELEMENT target_range_condition (#PCDATA)>
<!ELEMENT src_path_condition (#PCDATA)>
<!ELEMENT src_path (#PCDATA)>
<!ELEMENT target_path_condition (#PCDATA)>
<!ELEMENT target_path (internal_link , internal_entity?)+>
<!ELEMENT internal_link (#PCDATA)>
<!ATTLIST internal_link  value_binding CDATA  #IMPLIED >
<!ELEMENT internal_entity (#PCDATA)>
<!ATTLIST internal_entity  id CDATA  #IMPLIED >
<!ELEMENT link_map (range_map , path_map)>
<!ELEMENT combined_links (link_map+)>
<!ATTLIST combined_links  joined_on   CDATA  #IMPLIED
                        compound_on CDATA  #IMPLIED >
```

Fig. 7. The schematic DTD of the mappings file

```
F if_condition

if_condition:
  if_condition AND if_condition
  | if_condition OR if_condition
  | NOT if_condition
  | condition

condition:
  entity ISA entity
  | entity EXISTS
  | entity = eterm

eterm:
  term| constant
```

Fig. 8. The mapping condition syntax

4. Examples

In order to fully understand the whole mapping schema several examples should be presented.

Case 3 Example

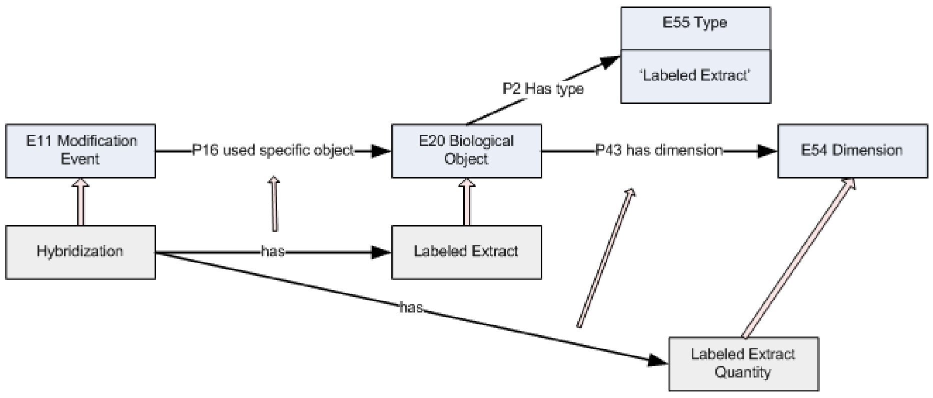

Fig. 9. Parallel to nested

The mappings that correspond to figure 9 are presented bellow in figure 11. It is common case, to have a single domain mapping and some range mappings so as we can see in the example we don't have to declare again the domain mapping. Moreover we can declare that the same instance of a class in a mapping rule appears in multiple mappings using the attribute "*joined_on*" of the "*combined_links*" tag. Note that when we have a value binding, we use the attribute "*value_binding*" of the tag "*int_entity*".

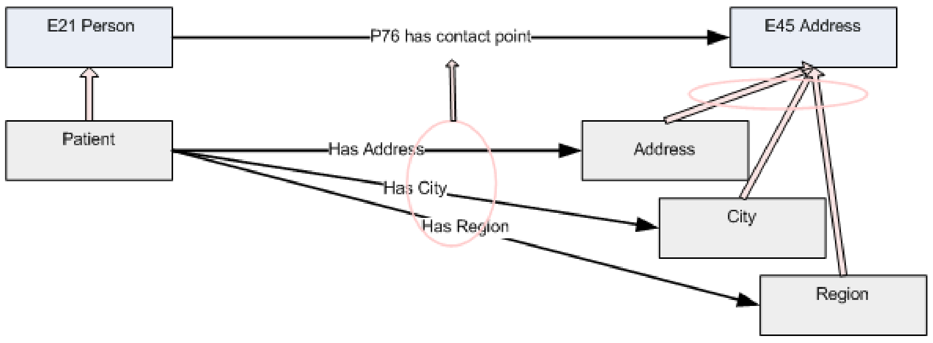

Fig. 10. Compound Contraction

```
<mapping>
      <map>
        <domain_map>
            <src_domain>Hybridization</src_domain>
            <target_domain >E11 Modification Event</ target_domain>
        </domain_map>

        <combined_links joined_on='x2'>
          <link_map>
                <range_map>
                        <src_range> Labelled Extract</src_range >
                        <target_range id='x2'>E20 Biological Object
                        </ target_range >
                </range_map>
                <path_map>
                        <src_path>has</src_path>
                        <target_path>
                                <int_link> P16 used specific object</int_link>
                        </target_path>
                                </path_map>
           </link_map>

           <link_map>
                <range_map>
                        <src_range> Labeled Extract Quantity</src_range >
                        < target_range>E54 Dimension</ target_range >
                </range_map>
                <path_map>
                        < target_path>
                                <int_link> P16 used specific object</int_link>
                                <int_entity  id='x2'  value_binding='has_type:  La-
                        beled Extract'>E20 Biological Object</int_entity>
                                <int_link> P43 has dimension </int_link>
                        </target_path>
                        <src_path>has</src_path>
                 </path_map>
           </link_map>
         </ combined _links>
      </map>
</mapping>
```

Fig. 11. The mappings of the first example

Case 2. Compound Contraction

The mappings that correspond to figure 10 are presented bellow in figure 12. In this example it is shown a case of compound contraction and the corresponding mappings produced.

```
<mapping>
      <map>
      <domain_map>
               <src_domain>Patient</src_domain>
               <target_domain >E21 Person</ target_domain>
      </domain_map>
        < combined _links compound_on='x3,x4,x5'>
         <link_map>
               <range_map>
                        <src_range id='x3'> Address</src_range >
                        < target_range>E45 Address </ target_range >
               </range_map>
               <path_map>
                        <src_path>has address</src_path>
                        <target_path>
                                 <int_link> P76 has contact point</int_link>
                        </target_path>
               </path_map>
          </link_map>
          <link_map>
               <range_map>
                        <src_range id='x4'> City</src_range >
                        < target_range>E45 Address </ target_range >
               </range_map>
               <path_map>
                        <src_path>has city</src_path>
                        <target_path>
                                 <int_link> P76 has contact point</int_link>
                        </target_path>
               </path_map>
          </link_map>

          <link_map>
               <range_map>
                        <src_range id='x5'> Region</src_range >
                        < target_range>E45 Address </ target_range >
               </range_map>
               <path_map>
                        <src_path>has region</src_path>
                        <target_path>
                                 <int_link> P76 has contact point</int_link>
                        </target_path>
               </path_map>
          </link_map>
        </ combined _links>
      </map>
      </mapping>
```

Fig. 12. The mappings of the second example

5. Conclusion

This document reflects the mapping process from our perspective (IT people) in order to make clear the whole process and the necessary mapping format to start building a mediator or data transformation system.

This mapping format is intended to be used by non-IT experts. It is designed, having in mind that doctor, clinicians, e.t.c, should be able to produce mappings of the data structures they deal with and understand, possibly assisted by an IT expert in the beginning. Therefore we tried to include the minimum information required in order to produce an efficient mapping.

Moreover, we expect that a graphical tool will be used. That tool should be available to make the whole process more intuitive, to ensure consistency between target and source schema and to help the experts with an appropriate visualizations for the result of those mappings.

We have to note that our mappings will be used in a Local as View mediation process and the mapping format should give enough specifications to an IT-expert to start building an integration algorithm without any help from the domain expert. Our plans for future work include the task of verification of our mapping mechanism. We have to verify that it is sufficient to produce complete mediation of data transformation algorithms without further input from the domain experts.

References

1. Avesani, P., Giunchiglia, F., Yatskevich, M. 2005., ``A Large Scale Taxonomy Mapping Evaluation", *Proceedings of the 4th International Semantic Web Conference (ISWC-05),* pp. 67-81.
2. Crofts, N, Doerr, M., Gill, T., Stead, S., Stiff, M. 2005., ``Definition of the CIDOC Conceptual Reference Model``
3. Gruber, T. 1993. ``Toward principles for the design of ontologies used for knowledge sharing. ", *International Workshop on Formal Ontology,* , 43 (5),pp. 907-928.
4. Kalfoglou, Y., Schorlemmer, M. 2003., ``Ontology Mapping: the State of the Art", *Knowledge Engineering Review*, 18 (1),pp. 1-31.